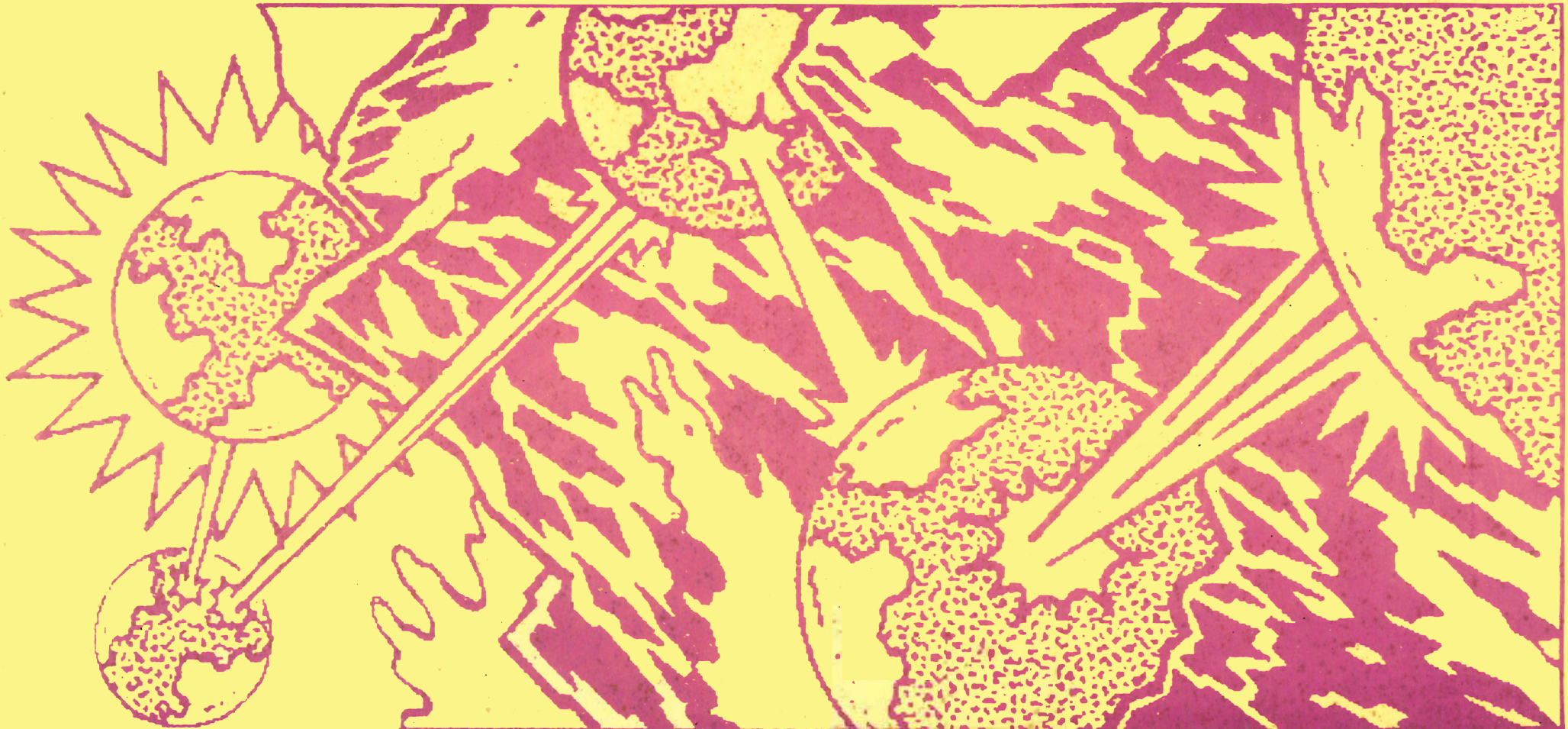
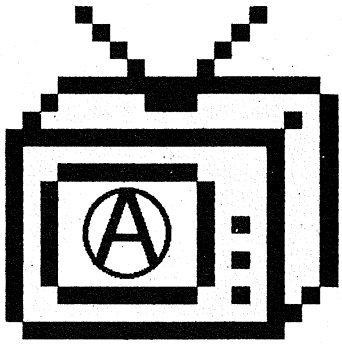
Message Sticks In Cyberspace



Will Kemp

Contents



Title and copyright notice
Free open licence to reproduce
Contents
Introduction

***** PART ONE - COMPUTERS *****

Chapter 1
1.1. What's in the box?

Chapter 2
2.1. How It Works

Chapter 3
3.1. Computer Basics
3.2. ASCII
3.3. Binary
3.4. Hexadecimal

Chapter 4
4.1. Communication
4.2. Ports
4.3. Connectors
4.4. Modems
4.5. Null Modems
4.6. Communications Software
4.7. Modem Speed
4.8. Modem Initialization
4.9. Calling a bulletin board

Chapter 5
5.1. Documentation
5.2. Manuals
5.3. Help
5.4. Keep records
5.5. Help is at hand

***** PART TWO - NETWORKS *****

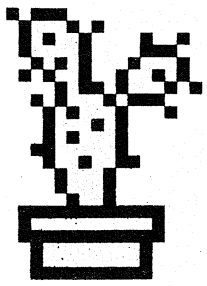
Chapter 6
6.1. Introduction To Networks
6.2. Networks For Anarchists
6.3. Different Types Of Network

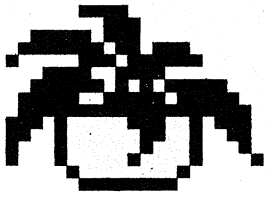
Chapter 7
7.1. An Anarchist BBS
7.2. Setting Up A BBS

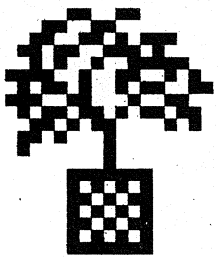
Chapter 8
8.1. Batch Networks
8.2. Setting up a feed in a batch net

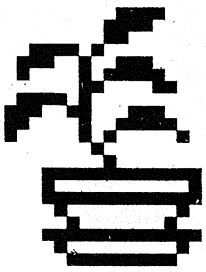
Chapter 9
9.1. Internet
9.2. What is it?
9.3. FTP
9.4. URLs
9.5. ARCHIE
9.6. Access to internet
9.7. Using internet

Chapter 10
10.1. Security
10.2. Network Privacy
10.3. BBS Security
10.4. False Names
10.5. Data Safety









Introduction

++++++++++++

The modern anarchist movement is torn between the current frenzy of computer hype and a natural and healthy luddite mentality. But we shouldn't let either of these stand in the way of capturing and using such an important means of communication.

In the last few years, the frantic pace of technological development has brought us to a situation where computer networks are one of the most important mediums of global communication. It's a medium we can't afford to ignore. It's cheap and relatively easy to use. It can save us a lot of work and it can help us to reach a wider range of people than ever before.

The main purpose of this booklet is an attempt to express the basics of computers in a practical and easy-to-understand way. It is orientated towards communications and is aimed at members of collectives and individuals involved in or interested in setting up an anarchist computer network.

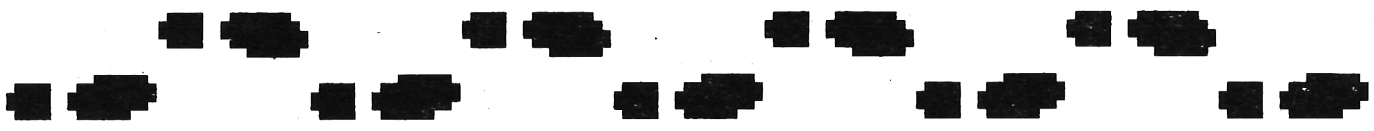
It's not meant to be a comprehensive guide to computers, but an introdution to get you to the point where you can set up and run a computer bulletin board service (BBS) and a network. If you have trouble understanding this shit, please don't worry. For one thing it's weird stuff and for another thing, i'm not very good at explaining it - nobody is!

If you start getting lost, toss it away and pick it up again tomorrow. It might make more sense then. Don't struggle with it, it'll never make any sense if you do!

It's better to read it through quickly a few times than to labour over it. Even if it doesn't make sense, at least you'll be familiar with some of the jargon. Also, try and read it in the order it comes in as it sort of explains itself as it goes along.

I chose the title of this book as a symbol of the down-to-earth use of this esoteric medium. And also as a small acknowledgement of the fact that i'm writing this on aboriginal land. Australia was colonized by the British ruling class in 1788. It's still a colony today.

I support the aboriginal people in their daily struggle to break the chains of colonization and hope that they too will find an international voice through computer networks.



***** PART ONE - COMPUTERS *****

Chapter One

++++++++++

1.1. What's in the box?

For the purposes of this guide, i use the word computer to mean personal computer. There are other types, but at this stage, we're not really interested in them.

The obvious components in a computer system are the screen, the keyboard and a box which makes a whirring noise when it's switched on.

The keyboard is a collection of electrical switches, the same as an electric typewriter and the screen is more or less the same as a television without its radio bits. But what's in the box is a complete mystery to most people.

Knowing about what's in the box is no more essential to using a computer than knowing about the engine is to driving a car. However, as with running a car, if you're trying to do it on the cheap it pays to know a bit about how it works.

The IBM-type pc is the Holden of the computer world. They're tough, easy to work on, parts are cheap and you can pick up an old, beaten up one for next to nothing.

From the outside, the average box, or system unit, has the following features: On the front, there's probably one or two floppy disk drive slots with little lights on them and at least one blank panel which may or may not have a little light on it. On some types of computers there are other bits and pieces here too, like switches etc. One one side, there may be a power on/off switch (although this could be on the back or - less likely - the front). On the back there's probably a whole collection of different connectors including one to plug in the screen, one for the keyboard and a mains lead.

With some computers (particularly the older ones) you must be careful taking off and putting on the cover as some of the cables inside can get snagged and damaged. They shouldn't be like this, but they often are.

The following description refers to the flat type of system unit (e.g. IBM XT, PC, AT etc), the upright type look a bit different, but it's only the way they're laid out.

UNPLUG THE COMPUTER FROM THE MAINS, take the cover off the system unit and have a look inside. For the time being, just look, there are some very sensitive components inside and you can damage them by touching them.

At the back left hand side, there's a series of long, multi-pin sockets, some of them with circuit boards standing up in them. These boards will also be fixed to the chassis with a screw. Some of the plugs on the back panel of the box are part of these circuit boards. This is basically where the computer does its communicating with the outside world.

At least one of these boards (known as option cards) will have cables coming off it which are connected to the floppy drives in the front of the box.

That card is the disk controller. This easily accessible array of option cards is one of the things that makes IBM-type computers so easy to work with. You can change them and add to them with a minimum of fuss. However, cards that work on one machine don't necessarily work on a different model. This is important!

Next to the array of option cards is a boxed-in bit with a grill on top. This is the power supply unit and the grill covers the cooling fan. It's the fan that makes the whirring noise when it's switched on. It's best to have as little to do with this bit as possible - and NEVER FUCK WITH IT WHEN THE POWER IS CONNECTED. However, if it fucks out, it's quite easy to replace the whole unit - but again, they're not all the same.

At the front of the box you'll see the floppy disk drives. These are similar to the tape deck in a cassette machine and are a mixture of electrical and mechanical bits. I

wouldn't recommend a beginner to try and work on them, but they're easy to take out and install if you're careful about getting the leads connected right. Again, though, not all floppy drives will work on all models - but this generally only applies to older models.

Somewhere around there too, there will be a sealed unit about the same size as a floppy drive and possibly connected to the same option card. This is the hard disk drive (HDD) and like the floppy drives it's easy to take it out and put a different one in.

At the bottom of the box is a large circuit board. This is known as the mother board and contains the processor and the main part of the computer's memory. The processor is the chip that runs the whole system and the mother board is roughly equivalent to the engine in a car - although it's considerably easier to take it out and put a new one in.

It's possible and relatively easy to dismantle the whole system box in a short time and replace any or all of the parts. You would normally only do this to repair or upgrade the computer. Don't be afraid of doing this sort of thing if you have to, but it's probably best to learn about it from someone with experience before you do.

Chapter Two

++++++++++

2.1. How It Works

I'm not going to go into the electronic technicalities of how computers work, but the basics of it are roughly as follows:

The processor is capable of manipulating data in a sort of numerical form in a range of different ways. However, to do this it has to be told exactly what to do to that data in a step-by-step way. Like if you had to tell someone exactly how to walk to another house: <<left foot forward, right foot forward, left foot forward..... turn right ninety degrees, left foot forward.....>> etc. It's tedious, but it's what the processor has to be told if it's going to do anything at all. A set of this type of instructions is called a program and these are stored in the computer's memory while it's doing it. When a program isn't being used, it can be stored on disk and only brought into the memory when it's needed.

Every computer system requires a whole set of programs to do the things it needs to do. These are known collectively as <<software>> and there's a vast ammount of it easily available. Most of it's shit!

The software that the computer uses to do what it does when it's basically not doing anything else is called the operating system (OS). Except when the power's off, computers are never idle. The processor's always running around and doing this and that, checking things and waiting for something to happen.

The OS keeps an eye on all of this as well as providing some basic functions like, moving, editing and deleting files, formatting disks, configuring the system how you want it etc.

Configuration is the process of setting up all the changeable things that affect the way the computer works. It ranges from basic things like screen colours, to much more complicated things related to the internal workings of the system. Configuring things seems to be an endless and irritating task sometimes and you seem to have to do it over and over again...

The capability of computers to be configured in an almost infinite variety of ways is what makes them so flexible and also what makes them an infinite pain in the fucking arse!

Basically, they're able to be set up in a million different ways that you're NEVER likely to want. This is partly because they were designed by a bunch of psychotic lunatics who had no idea at all what they were doing, and partly because the computers themselves and all the bits that connect to them were all designed by companies who had their own ideas about how things should be done and their own greedy, selfish, commercial reasons for wanting to make everything as incompatible with everything else as possible. It's also because they are used in a lot of different circumstances for lots of different things.

Instead of having one box that's good at file handling, one box that's good at communications, one box that's good for running accounts programs etc, etc, (like a hi-fi system) we've ended up with one box that can do almost anything if you set it up in the right way. There's obvious advantages to this, but the big disadvantage is you've virtually got to know almost as much about the things you *don't* want to do as the things you *do* want to do.

This isn't strictly true, but as you'll discover if you have to set up a system like a bulletin board from scratch, you have to poke around with an endless number of configurable things to get it working right.

Most of this fucking around really shouldn't be necessary, but because of the limitations and incompatibilities of the software it has to be done. There are two main factors which influence this - firstly, computer programmers really aren't very good at what they do and secondly, it's their interests to maintain the highest of mystification possible. If no-one else understands the things you do, it gives you a lot of status and power (and, like priests, it gives you a direct line to god!)

But don't let this nonsense put you off programmers are usually so thick, that if

they can work out what's going on, anyone can!

Anyway, back to operating systems...

The two main OSs you're likely to encounter on IBM-type systems are DOS (stands for Disk Operating System) and Windows.

DOS is extremely stupid and doesn't give many clues on what to do if you don't know anything about it, but it's efficient (that is it works relatively fast and doesn't take up a lot of space in the computer).

Windows is much smarter and infinitely easier to use if you don't know anything about it than DOS is. But it's heaps less efficient.

If you're setting up a bulletin board system, you may be able to use Windows for some things but you will inevitably have to become familiar with DOS.

Other OSs you may run into are OS/2 (similar to Windows, but much better), Unix (used for most Internet stuff - as easy to use as DOS), Linux (a freeware system, compatible with Unix), CP/M (dunno anything about it and you're not likely to need to either!) and (not on IBMs of course) Macintosh operating system (this is similar to windows, but a whole other story).

Windows, OS/2, Unix and Linux are all capable of what's called <<multi-tasking>>. This means you can run more than one program at the same time. You will need to be able to do this if you want to run a multi-line BBS (see BBS section). DOS can't do this. However, DOS *can* run on even the most primitive PCs

Chapter Three

++++++++++++

3.1. Computer Basics

At this point, i'm afraid we have to get a little more technical. No, STOP! Don't start turning the page! It won't be that bad! However, some of these concepts are a bit strange at first, but don't worry if you don't pick it up straight away, just aim at becoming familiar with the jargon. It doesn't really matter whether you understand it or not, you can always come back to it and have another look later, it might mean more when you've got a more solid context to put it into.

The basic unit of information that computers work with is called a <<Binary digIT>> - almost always known as a <<bit>>. One bit on it's own can represent the values '0' or '1' (or 'off' or 'on') (or 'yes' or 'no'). For pretty well all purposes they are referred to as '0' and '1'.

Inside the computer these values are represented by two different electrical voltages. But mostly, when the computer takes in or puts out data, they are represented in a different way. This is because these electrical voltages don't travel very far down wires, you can't put them onto disks and of course they mean fuck all to humans!

So for sending down phone lines etc, they have to be converted into sounds similar to tone dialling in a telephone. For storing on disks, which are basically the same material as audio tape, they have to be converted into something similar to sound, but higher pitched. And of course, for communicating with people they get converted to and from text (via keyboard, screen and printer), graphics or sounds.

When the computer is working with text (letters, numbers etc) it's represented by groups of bits. Each character has its own code and it's these codes that the computer actually manipulates.

3.2. ASCII

The most common system of character codes is called ASCII (this stands for American Standard Code for Information Interchange). There are others, but you're pretty unlikely to ever come across them.

Ascii codes are made up of 7 bits. There are 128 possible combinations of 7 bits. Therefore, standard ascii has 128 characters. This includes 10 digits (0-9), 26 capital letters and 26 lower case letters, as well as the other normal symbols you find on a keyboard, e.g. &,$,!,@ etc. The rest of the character set consists of control characters which are used for special (and often different) purposes by software.

It's important to bear in mind that these characters have no significance whatever to the hardware, which always works with the binary codes for them and never sees anything other than collections of 1s and 0s.

The next most basic unit of information that computers work with is called a <<byte>>. A byte is a group of 8 bits. there are 256 possible combinations of 8 bits. A byte can be viewed as a number and its maximum value can be 255 (as one of the combinations equals zero). That is a byte can have any value from 0-255.

As the byte is a pretty well universal form of data, this means that if you use a byte for each character (which is normal) you *could* have 256 characters instead of the 128 which ascii has. IBM-type computers make use of this possibility by having an extended form of ascii which allows an extra 128 characters. However, as not all computers use the whole 256 characters and the ones that do use the extra 128 in different ways, this extended code isn't a lot of use for communicating with other computers.

The extra 128 characters are known as IBM codes and it's best to avoid using them in a network setup.

It's important to remember that bytes are used to represent other things than characters too. Computer programs are

stored in exactly the same format. Only here, the bytes represent the codes for processor operations and addresses of data and software. As well as this, they can also be used to represent their numerical value for mathematical operations within the computer. It's also possible to allocate an on/off meaning to each bit and this could be used to convey all sorts of things from one bit of software to another.

So if you try and look at a software file as if it was a text file, you'll get completely meaningless nonsense. Or if you try and get the computer to run a text file as if it was software, completely unpredictable things will happen and the computer will almost certainly crash.

In the system used by DOS for naming files, this is taken care of to a certain extent. DOS file names have the form:

<filename>.<extension>

e.g: thisfile.txt

'thisfile' is its name and 'txt' (the extension) indicates what type of file it is. txt = text (that is, straight ascii usually).

other file types inclued:

.com = program file

.exe = program file

.bat = batch file (instructions for DOS to use almost like a program)

.bin = binary (could be program or data)

.doc = document (usually word processor format)

.asc – ascii

.sys = system files

.zip = file compressed with PKZIP compression

.arj = ditto - ARJ compression

.lha = ditto - LHARC compression etc.

The last three will need to be uncompressed with the correct uncompression program.

For text files it's always best to use an extension of .txt or .asc , although you will find text files with all sorts of extensions.

3.3. Binary

So what is this <<binary>> shit, anyway? Well, in decimal, every digit represents a value from 0 to 9 and every time you move one digit to the left, its value is multiplied by 10. In binary, every digit represents the value 0 or 1 and every time you move one digit to the left its value is multiplied by 2.

In decimal notation this means the value of the digits (from right to left) is as follows:

1s, 10s, 100s, 1000s, 10000s etc

In binary notation the values are:

1s, 2s, 4s, 8s, 16s, 32s, 64s, 128s etc

Put the right way round, that means:

128, 64, 32, 16, 8, 4, 2, 1

That is what the eight bits of a byte represent.

Examples

00000001 = 1 00000010 = 2
00000011 = 3 00000100 = 4
10000000 = 128 10000001 = 129
10000010 = 130 10000011 = 131 etc...

Any binary value can be converted to decimal by working out and adding up the values of all the 1s. e.g.:

10110010 = 128+32+16+2 = 178

I'm afraid that's as good as i can explain it. If it doesn't make sense, you'll have to keep trying or get someone else to explain it differently. It's important to understand this stuff, although it's possible to get by without it. The more of these basics you understand, the easier it will be to sort out problems later.

And if you thought that was weird, have a look at this:

3.4 Hexadecimal

Because of the need to work with these binary values and because binary numbers are so long and hard to remember and express, a shorthand notation is used.

This is called hexadecimal (because each digit can represent 16 different values).

In the way that in decimal each digit can have values from 0-9, in hexadecimal each digit can have a value from 0-15. The digits from 10-15 are represented by the letters A-F. So the complete range of hex digits is:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f.

Right, what's all this got to do with binary? Well, if you group binary values in chunks of four bits, each four bits can have 16 different combinations and therefore numerically can represent values from 0-15. The same as each hex digit! Hey!

Therefore, a byte can either be represented as 8 binary digits or 2 hex digits. Or, of course, 3 decimal digits. As hex is more handy than binary, while still being really easy to convert to binary, and decimal is really difficult to convert to binary, hex is very commonly used in all fields of computing.

Here are the binary and decimal equivalents of the 16 hex digits:

binary	hex	dec
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

So 10110010 = B2 (hex) = 178 (decimal)

Chapter Four

++++++++++++

4.1. Communication

The main form of communications that concerns us in setting up bulletin boards and networks is direct communication between two computers. The two methods of connecting computers together that we'll consider here are (1) via the telephone system and (2) via a simple cable between two physically close computers.

There are two basic modes of communications that computers use:
these are called 'serial' and 'parallel'. In serial communication the eight bits of each byte are sent down a single wire, one after the other. In parallel communication, the eight bits are sent down eight separate wires at the same time. Parallel communication is obviously considerably faster, but serial communication is much easier to do and can be done over a phone line.

4.2. Ports

On the back of the computer, there will almost certainly be some multi-pin sockets. Each of these is part of an option card that's plugged into the mother board. There's likely to be at least one parallel port and one serial port. A port is a connection through the which the computer communicates with the outside world.

Printers usually plug into a parallel port and mice use serial ports (that is, if they can't find a hole big enough to get inside and chew up the wires!) The one that we'll be using is the serial port.

The computer you're using is almost certain to have a serial port, however if it doesn't you can buy a <<serial card>> quite cheaply and plug it into one of the connectors on the mother board. If you install one of these you will have to configure it in a way that will work with your computer. The supplier can explain this. It can generally be plugged into any vacant slot on the mother board.

4.3. Connectors

There are several types of connector you might find on the back of your computer. Some will be <<male>> connectors - that is, they'll have little metal pins which fit into holes on the (<<female>>) plug. Others will be <<female>> - that is, they have little holes which pins on the (<<male>>) plug fit into. There will probably be different sized ones too. The size of a connecter is referred to by the number of pins it has on it.

Serial ports are usually either 25 pin or 9 pin male connectors. Parallel ports are usually 25 pin female connectors. So to connect something to a serial port, you'll need a cable with a 25 pin (or 9 pin) female plug on one end and the correct plug for whatever you're connecting to it on the other end.

4.4 Modems

If you're going to connect to a computer that's not in the same room, you'll need to use a modem. This is a unit that plugs into a computer on one side and into the phone line on the other side.

As explained in chapter three, the signals that computers use won't travel very far down wires and therefore have to be converted into audio signals, which do travel well through long wires. This is what a modem does.

<<Modem>> is short for MOdulator/DEModulator. Modulation is the process of converting digital signals into sound signals.

As well as converting the data into a form in which it can be sent down a phone line, modern modems are able to dial phone numbers and answer incoming calls automatically.

There are two types of modem: internal and external. They both do exactly the same thing. The difference is that one goes inside the computer - it is in the form of an

option card and plugs into the mother board - and the other is in a box of its own and must be connected to a serial port via a modem cable.

Both types of modem have their advantages. Internal ones are considerably cheaper and don't require a separate serial card, however they can't necessarily be used with other types of computer. External modems really only have the advantage of being compatible with all types of computers (given the right cable) and they can be easily disconnected and used somewhere else. Either way, they are easy to install and operate.

4.5. Null Modems

If you have two computers you want to connect up in the same room, you can connect them together with a special cable called a <<null modem>>.

A null modem is simply a cable that's configured in a way that connects the serial ports of the two computers together in the right way for them to work properly. (The connector on a modem is wired up so that all the pins on the computer's serial port are connected to the same pins on the modem. This won't work between two computers.)

4.6. Communications Software

Once you've plugged the modem in, you've got to have some way of making it do something. This isn't something the operating system usually does - not DOS, anyway. For this you need some communications software. What this software does is set the serial port and the modem up in the right way and do all the dialling or call answering stuff as well as perform other functions needed for effective communication.

At the moment, we'll only consider calling out to another computer. The job of taking incoming calls usually falls to Bulletin Board software and we'll talk about that later.

So, say you want to use your computer and phone line to call up a bulletin board. What do you have to do to get it happening?

Firstly, the software must be configured to do what you want it to. This is usually not very difficult. The comms prog will have a setup menu to do it. All software packages have a default way of setting themselves up and this is usually good enough to start with. When you get familiar with it, you may want to change some of the ways it's set up, but it probably won't be totally necessary.

One thing you must set up though, is the way the software talks to the modem. Firstly, you must make sure it's using the right communications port. The computer has its internal ways of identifying different ports and it's possible to have at least four different serial ports. When the serial card was installed, it would have been configured to be a particular port. There's a good chance that this will be COM1, but not necessarily. You must either know this or fuck around trying different things till it works!

Once you've got the right COM port, you have to set the speed that it's going to talk to the modem at. For this, you'll need the modem manual. It will probably be best to set it at the modem's maximum speed, however, with some modems, it's better to have it a lot faster than that.

4.7. Modem Speed

Right, here we go! There are two speeds associated with modems - 1) the speed it sends and receives data down the phone line and 2) the speed it talks to the computer at.

Now what exactly do we mean by speed? Basically, it's how many BITs are transmitted in a second. But it's not exactly that! Do you really want to know?

Well, i'd better tell you anyway...

Data transmission speed is measured in bits per second (bps). Another word for bps is 'baud'. 1 baud = 1bps. However, it's not quite that simple as the communications hardware needs to use some of those bits

for its own internal control purposes. Therefore if two computers are talking at 2400 baud, you won't get 2400 bits per second of usable data. What you will get depends on what's called the 'protocol' you're using. Protocol is the way the computers control the communication between them and check for errors etc.

With most of the communication we're likely to be involved in, the thing that interests us most is how many characters we're getting per second. And even this isn't really essential knowledge. However, it helps to be familiar with this stuff so you know what other people are on about when they refer to it.

When you set up your communications software, you'll see a section where you can specify 'parity', 'data bits' and 'stop bits'. These will probably be set to 'N', '8' and '1' by default and this is what we're likely to use all the time.

Parity is a primitive form of error checking which more or less comes from adding up all the bits in a byte in a way that leaves you with just one bit (0 or 1). This is done before the data's sent and then again by the computer receiving it and the parity bits should come out the same both times. However this is a pretty unreliable form of error checking and it's rarely used nowadays. The 'N' stands for 'none' (i.e. no parity checking). The other possibilities are Odd, Even, Mark and Space, but if you're interested in them you can find out about them yourself!

The '8' stands for 8 data bits to a byte. The other possibility is 7 - which leaves one spare bit for parity. This is why the ASCII code is 7 bits (to leave a spare bit for parity - which isn't used much anymore, what a waste!)

The '1' means 1 stop bit. Stop bits are what the hardware uses to synchronize communications with the other computer (or with the modem). The alternative is 2. As well as either one or two stop bits being added to the end of each byte, one 'start bit' is added at the beginning for the same reason.

So obviously, for two computers to talk properly, they must both use the same data format (e.g. N.8.1).

How does this affect transmission speeds? Well, if you've got 8 bits of data you want and 2 bits that the computer needs (1 start & 1 stop bit), you're not going to get 2400 usable bits per second out of 2400 baud. What you do get is 1920 bps, which is 240 characters per second (cps), rather than the 300 cps you might expect. Normally, the baud rate is used to refer to how fast data is going down the line, while bps refers to how many usable bits you're getting.

What i've said above is true for simple computer-computer or computer-modem communication but isn't strictly true when it comes to modems. The baud rate of a modem-modem connection is a totally different thing to the bit per second rate. This is because, baud really refers to how often the signal can change and every signal change in a modem running at over 300 bps signifies more than one bit. (Erm... I don't want to get too technical about this, but i feel it does need a bit of clarification.)

For example, at some speeds a modem will communicate four bits with one signal change. That means each baud = 4 bps. Etc etc...

As i said, you really don't need to use this knowledge, but it does come up quite often in discussions on BBSs, and it also helps clarify some of the settings of your comms software. Don't worry if you don't fully understand it.

4.8. Modem Initialization

The other thing that must be set up properly before your communications software will work properly, is the way the modem's configured.

Most modems are what's known as <<Hayes compatible>>. This means they accept commands from the computer in a standard form. You'll need to refer to your modem manual to find out about the commands you need to get it working in the right way.

Some simple options include whether or not the modem automatically answers

incoming calls, whether the speaker in the modem lets you hear it dialling or not, whether it uses tone or pulse dialling, etc. There will also be a command for it to dial a number.

These commands take the form of 'AT' at the beginning, followed by letter and number codes for the functions it is to perform.

For example, the command to dial 03-123 4567 would be:

ATD031234567<CR>

The <CR> means carriage return and is represented in different ways by different comms s/ware. check the manual on this. However, your comms s/ware will almost certainly allow you to dial a number more easily than that.

TELIX is the name of the comms program i use most and it has a phone directory which you add numbers to with simple commands. You can then dial them by placing a cursor at the entry you want and typing 'd'.

But you may have trouble with the modem not working properly. This usually comes from it not being 'initialized' correctly.

Initialization is something the comms s/ware does when you first start it up. Amongst other things, it sends a command string to the modem to configure it correctly.

The comms s/ware will come with a default initialization string ('string' just means a group of characters, e.g.: ATD031234567).

The default initialization string may work, but it probably won't as all types of modems seem to be slightly different in the way they're set up. If you have problems, you'll have to read your modem manual and either try and find an initialization string the manufacturers reccommend or make your own one from the information in the manual. This is always one of the first things to check if your modem is doing weird things.

You can set up the initialization string in the comms program's setup menu.

4.9. Calling a bulletin board

Right, well there's all sorts of stumbling blocks on the way, but let's assume you've got the whole system sussed out and working. What now?

The obvious way to test it all out is to call up another computer and use it to talk to that computer with. There are lots of public access computer systems, known as Bulletin Board Systems (BBSs). These are mainly run by single individuals or small groups as a community service. Access is generally free and various forms of participation are supported.

The main functions of BBSs are message exchange, file exchange and game playing. For a fuller description of BBSs check out the chapter on that subject in part two.

So the best way to check out your comms setup is calling up a BBS. How you find the phone number of your first BBS, i'm buggered if i know! Once you've found one, you can usually find all the rest in one way or another. In australia, there's a national BBS list which is circulated amongst BBSs, but i don't know how you'd get it without already knowing one. Also, most BBSs carry a few numbers of other BBSs of a similar mentality. Occasionally you'll see an article in a newspaper on the subject which will have a few numbers and i've never looked at a computer magazine, but i guess they probably have that sort of thing too.

Once you've got your BBS phone number, you can enter it in the dialling directory of your comms program. And then try dialling it.

And...

Well, there could be several reasons for that screenfull of meaningless garbage - like the protocol's not set right (N,8,1, etc), modem speed's wrong, computer speed's wrong, serial port card's fucked out.....

What?! You haven't got a screenfull of meaningless garbage?!?! Far out! Well, there ya go, life ain't all bad is it?

So what will you see when your computer's connected to the BBS?

Sometimes, nothing at all for a while, as the BBS computer loads up its comms s/ware and sorts itself out. If it seems to be taking a long time to happen, hitting <ENTER> or <ESC> once or twice might spark it off.

And then you get maybe a title screen with the BBS name etc, or maybe just a prompt asking for your name (some systems ask for LOGIN: - this is the same thing).

When you enter your name, the system will recognize that it doesn't know you (on your first call, that is) and ask you a few questions. What these are depends on the individual sysop (the person that runs the BBS - short for SYStems OPerator). But one thing it will ask you to do is choose a password. This is a word you'll use every time you log on to (call up) this BBS so it knows you're really who you say you are. It should be at least 6 characters and something that won't be an easy guess for someone else. You'll be able to change this any time you feel like it.

You'll also probably get asked questions on the setup of your computer, like screen size etc. Usually, hitting <ENTER> will select the default value and this will be right - again, you can change all this sort of stuff later.

Mostly, finding your way around a BBS is easy. You should be presented with menus and clear choices all the way and maybe even

<<help>> commands for when you get stuck.

The best start is to read through some of the messages on the first message area you come to. Depending on the particular software the BBS is using, this will be simple and straightforward, or slightly confusing. Either way, it just takes a bit of getting used to. Don't panic! It will all become clear in the end! Maybe!

One command that needs explaining is the 'Logoff' command. This means <<end this call to this BBS>>.

If the first bulletin board you call isn't very interesting, that's normal. There's lots of them, they all cater for different types of people and there's always one which you'll find interesting. Just keep groping around and you'll find it.

After your session on the BBS, check your modem has hung up the phone right. Sometimes they don't - it's another case of getting it configured right...

So if everything went right, you're ready to take on the freaky world of computer communications. But watch out! BBSing can be addictive. Some people spend hours every day reading and writing messages, playing games etc on BBSs...

Chapter Five

++++++++++++

5.1. Documentation

Most computer software comes with some form of documentation. This is usually a text file that comes with the program files (s/ware you get from a BBS) or a printed manual (if you buy it from a shop). Generally speaking the standard of this documentation is very low! It often makes you wonder how they managed to get it to work at all if the logic they've used in writing the programs is the same as what they've used to organize the manual!

5.2. Manuals

With manuals that come in the form of a computer file, it's pretty essential to print them out onto paper so you can refer to them at the same time as you're fucking around trying to get the program to work. It's impossible to do this *and* try and read the manual on the screen too! Anyway, it's much harder to find what you want in a long text file than it is on sheets of paper.

Reading manuals is a skill that takes a certain amount of practice to get good at. It shouldn't be really, but they really are incredibly bad! It all helps to maintain the mystique which the whole game is shrouded in.

There are two approaches to manuals - and sometimes one is better than the other. You can sit down and read it from cover to cover before you start, so you know what you're doing from the beginning. Or you can work through bits you need as you need them and try and get the program working before you've really read the manual - using the documentation where you run into problems.

Neither way is fully satisfactory. And both ways together is more or less the best way to do it. However, you've got to be a particular kind of weirdo to be able to sit down and read a computer manual through (yes, *i* can, but i'm weird!) They're usually so badly written and baffling that you can't follow it anyway. And of course until you've had some experience of using the software, a lot of it doesn't mean much either.

I would recommend trying to read the manual first - at least the preliminary bits. But don't be put off if you can't read it right through - almost nobody can! Do try and get familiar with the way it's organized so you can find things you need when you're looking for them (chuckle!) (You'll be lucky!) And also get familiar with some of the terms it uses.

5.3. Help

The other form of documentation is the help functions that are built into the program itself.

Most software has some form of online help. It may be in the form of menus which guide you through the operation of it, or there may be a <<help>> command of some sort, which displays information on the screen. This stuff is very useful and is sometimes more help than the manual. Get to know your way around these features of the software as soon as you can.

5.4. Keep records

Another aspect of documentation is the records *you* keep. If i had a dollar for every time i had to solve the same problem twice, i'd be very rich now! And how many of those times have i kept a record of what the problem was and how i solved it? Not many, sadly! There's nothing more frustrating than grappling with a problem you've solved before and not being able to remember how you sorted it out last time! You'd be amazed how often this happens.

It's important to keep a log of all this stuff. It's easy to do - just write it in a text file as it happens. This file can be shared with other people doing similar things and for training. It will make everyone's lives

easier.

5.5. Help is at hand

When all else fails and you're completely stumped, don't despair, help is never far away in the computer world - so long as your comms s/ware's working, that is! Every BBS is frequented by people using computers. Most of them have solved a few problems in the past and most of them are more than happy to help someone out with a bit of advice.

A lot of BBSs have a <<technical>> message area where these things are discussed. Otherwise, a quick message in the general message area should bring a few helpful responses.

Someone out there somewhere will know how to sort it out. If you don't get the help you need straight away, try another BBS or a different message area. This way you'll definitely find a solution.

BOSSES AND BOSSED
HAVE THE KIND OF RELATIONSHIP
WHICH CAUSES MISERY
AT EVERY LEVEL.

***** PART TWO - NETWORKS *****

Chapter 6

++++++++++

6.1. Introduction To Networks

The term <<network>> refers to a situation where two or more computers are linked together in such a way that they can pass information between them.

The simplest form of network is probably two computer users swapping floppy disks with files on them. This way information is being transferred from one computer to the other by a human medium. Although sending floppy disks through the mail may well come into the networks we'll be building, it doesn't really come under the heading of computer networks. However the principle is basically the same.

The usual meaning of the term <<computer network>> is two or more computers exchanging information via and electronic medium (such as a telephone line).

There are two basic types of computer network, which i'll call <<batch>> and <<realtime>>. In a realtime network, the computers involved are permanently connected together and pass information between them as and when it's needed. In a batch network, the computers call each other up (usually by phone) at regular intervals (maybe once a day) and swap all the traffic that's accumulated since they last called.

Both types of network may be important to us as a medium for anarchist communication.

Realtime networks also may be divided into two types - local area networks (LANs) and wide area networks (WANs). A local area network refers to computers in the same building or maybe adjoining buildings linked by cables which are fairly short. Wide area networks involve computers that are too far apart to be able to just run a wire between them. They could be on the other side of town, or the other side of the world. Internet is probably the best-known of this type of net. These networks use modems to send data over long distances.

6.2. Networks For Anarchists

So what does all this mean to us?

Well, in my opinion, one of the main goals of the anarchist movement is communication. Communication of ideas in the hope of bringing about political change, communication between groups to share work and help with organizing, communication between individuals to help get together in groups and communication to help us feel we're not the only people in the world with these political beliefs.

Computer networks can provide a cheap and easy means of both person-to-person and mass communication. Admittedly, you need a computer a modem and a phone line to participate, but then you need a television to watch t.v., a radio to listen to radio broadcasts, a telephone to receive phone calls - or, of course, none of these if they don't interest you.

However, i'd like to see anarchist computer networks being made available to people who don't own computers. This can be done by setting up public access terminals in community centres, bookshops, infoshops etc. And also by printing stuff out and distributing it on paper.

Through computer networks, we can automate the propagation of news and information across a town, across a country, or around the world. It can be done relatively cheaply and it can make it a lot easier.

Instead of writing a bulletin, typing it, photocopying or printing it, putting it into envelopes, writing the addresses on them, buying stamps etc, how much easier would it be if you could just type it once into the computer, and: bang! off it goes? You don't have to worry any more about it, by tomorrow it will be all round the world!

Of course it's not quite that simple,

the people you want to get information to still have to have access to the network. But that's the basic principle of it. A lot of communication that we currently have to do in a laborious way - or, more likely, don't do at all - could be done with computer networks.

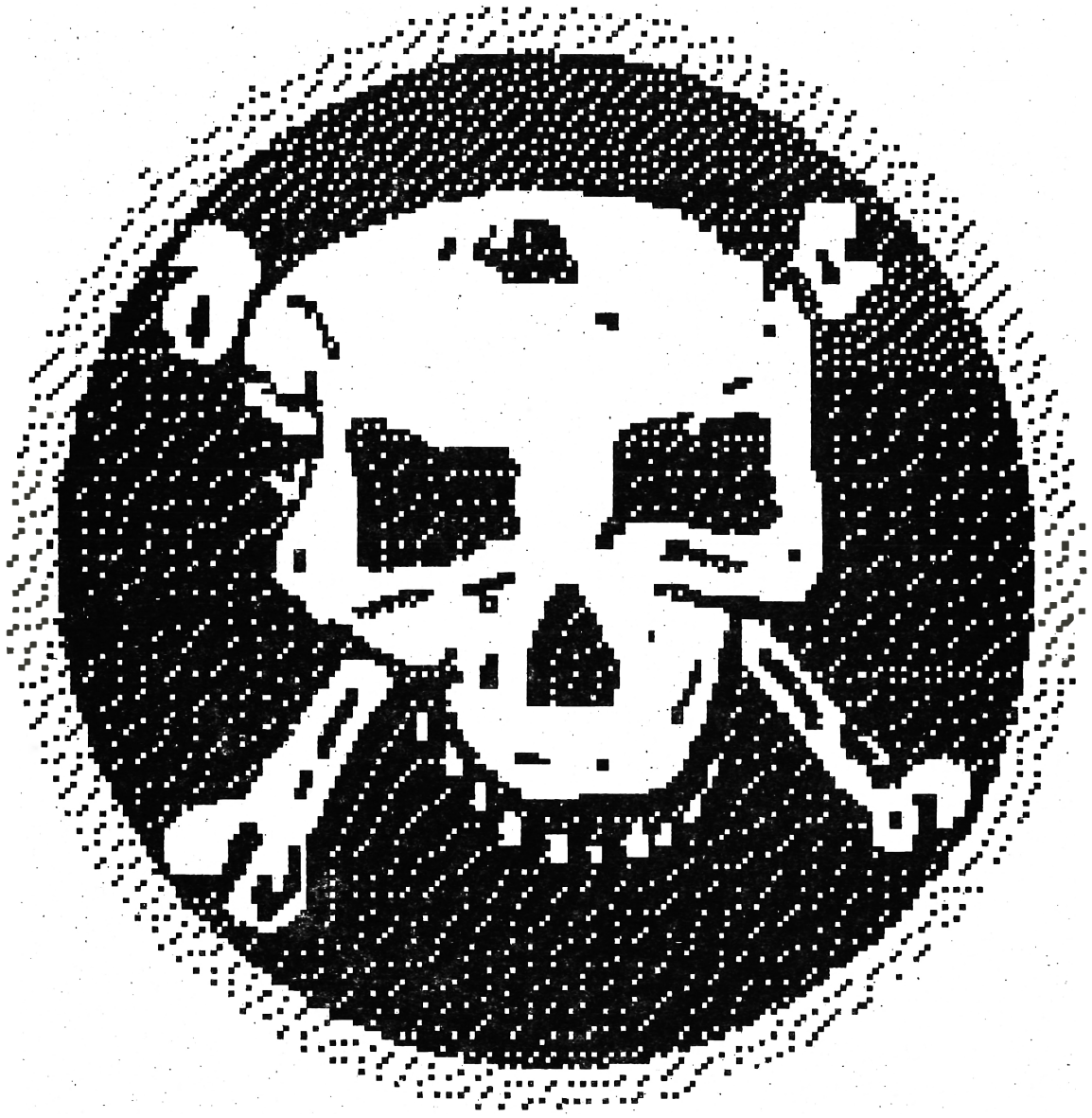
6.3. Different Types Of Network

The two main types of network which are of interest to us in this context are fidonet type networks and internet.

Fido-type nets are made up of a number of bulletin boards, which could be anywhere in the world, linked by regular exchange of messages over dial-up phone lines.

Internet consists of thousands of computers of all shapes and sizes, all over the world, linked mainly by permanent (sometimes extremely high speed) lines twenty four hours a day. There are also a large number of computers linked to internet by regular dial-up connections similar to fido-type nets.

Obviously having a computer linked permanently to a network is much more expensive than having it dial up once a day. And although we'll touch on realtime internet links, we're mainly interested in batch-type networking.



Chapter 7

++++++++++

7.1. An Anarchist BBS

I'd say the most useful tool in computerized anarchist communication is the bulletin board. You can load a bbs up with text files of all your favourite propaganda which then becomes available to anyone calling in. You can use it an a means of public access to your favourite anarchist computer network - or even to internet allowing anarchists to communicate with each other globally for the price of a local phone call.

Callers can read articles, books etc on-line, or they can transfer them to their own computer and read them offline and print them out. They can read and post public messages about actions and events locally and internationally. They can take part in message-based discussions with people locally or anywhere the net reaches to. They can also send personal mail to anyone who also has access to the same net. They can get software which they can use on their own computer. If whoever set up the bbs was that way inclined, they could also play games on-line.

This of course would all be wonderful - if there were lots of anarchists with computers and modems. Which there aren't! More and more people do seem to be getting their own computers, but for most people in the european world this technology is still psychologically, if not financially, out of reach. Of course, for most people in the non-european world, it's not even a remote possibility.

However, computers make far better communal tools than they do private ones. Like most things, i suppose, your personal computer is likely to spend most of its life sitting idle (unless, of course, you're a demented computer geek!) But if it was to find its way into a communal space, where lots of people had access to it - and if it was set up in such a way that you didn't have to have a degree in nuclear physics to operate it - it could become a public access gateway to a mass medium.

Of course, it's not necessary for people to go anywhere near a computer to get access to news and information from an international computer network. Every time you read a national newspaper you're doing exactly that.

A good way to use computer networks for anarchist communication could be as a transport medium, bringing material into your desktop publishing program. With a minimum ammount of work, material gathered from the network could be edited, typeset and printed out in newsletter or magazine form. An international anarchist weekly newspaper, being published locally all round the world and with an international editorial collective is a very real possibility.

One thing computer communication does give us is the possibility to rival mainstream capitalist media in this way. It's all there, available and it could be done now.

So that's the <<why>> of anarchist bbss, what about the <<how>>?

7.2. Setting Up A BBS

If you read part one and understood it, especially if you've managed to set up a modem and use it to call a bbs, you're quite capable of setting up and running your own bbs. If the above statement doesn't apply to you, you're probably still capable of setting up and running a bbs, but you might need a bit of help (shit! we all do!). If you're definitely *not* capable of it (and i suppose some people might not be...), you almost certainly know someone who is.

If you're interested in setting up a bbs, you should start by spending as much time as you can calling up some of the bbss in your area and getting the hang of how to use them. You'll also begin to get an idea of how they work and, hopefully, how networks work too.

Ask lots of questions... Some bbs users like to look down on people who don't know much, but most of them are more than willing to pass on everything they

know to someone else. Spend enough time lurking around on bbss and you'll learn everything you need to know.

However, it's possible that you don't have access to local bbss, in which case, i'd better explain the whole thing from scratch. Of course, i can't explain everything, and i can't anticipate all the problems you're likely to have (probably not even a hundredth of them - no i'm not joking!), so you'll end up having to work out most of it as you go along.

All the software should come with documentation (manuals etc), and although they'll be hard to understand and will omit to tell you the most important things, they're mainly what you should use when you set up that particular piece of software.

If you've checked out a few different bbss, you'll probably have discovered that bbs software comes in several different flavours. A few examples off bbs software names are: SuperBBS, QuickBBS, Waffle, Wildcat, Maximus, etc. They all have different approaches, but they basically all do the same thing.

The basic functions of a bbs are as follows:

1. it must answer the phone

2. it must ask callers for their name and a password so you can control access to the bbs itself and also to different areas and services within the bbs.

3. it must allow users to read and write messages so they can communicate with each other.

4. it must allow them access to certain file areas where you can put material you want to make publicly available.

5. it must allow them to transfer files between the bbs and their computer (in both directions, if you want public contributions to your file areas).

6. it should allow users to transfer mail from the bbs to their computer so they can read it offline - particularly where local phone calls are charged by time. It should also allow them to transfer the replies back from their own computer to the bbs later on.

7. it should allow you to install <<plug-in>> bits of software so you can run games and other things that don't come with the standard bbs. offline mail (see 6.) may be one of these.

8. it should allow your computer to automatically make calls to other computers at pre-arranged times, to transfer messages around a network. It should also be able to accept calls from other computers doing the same thing. This last part may or may not be part of the main bbs software. In some setups, it's done by a separate program.

All bbs software packages are capable of doing all the above. If they're not, they're useless.

Chapter Eight
++++++++++++

8.1. Batch Networks

There are two different protocols used for batch networks. Fido and UUCP. Fido comes from Fidonet, which was the first bbs network. It was originally a network of DEC pdp8s which, for some reason, were known as <<fidos>>. UUCP stands for <<unix to unix copy protocol>>. Unix is an operating system that was originally used on mainframes and mini-computers and UUCP is what unix uses for batch networking.

The term <<mainframe>> refers to large computers that were really all there were in the '60s. Minicomputers like the pdp8 and pdp11 began to take over from them in the '70s, and then of course, in the '80s along came the micro and turned the whole business of computing on its head.

There are, however, still a lot of mainframes and minis in use around the world.

There are two fundamental differences between fido and uucp. Firstly, the protocol used by the computers to conduct the passing of data is different. Secondly the form of message addressing is different.

Fido uses numerical addresses which look something like:

2:317/0

while uucp uses word addresses in internet form, like:

will@byteback.apana.org.au

uucp also uses a form of addressing which specifies the route a message must take to get where it's going. This is known as a <<bang path>> (bang = !) and looks like:

troll.apana.org.au!byteback!will

(which is the same address as the one above, but tells you a bit more about how it gets there - troll is the apana computer that byteback connects to internet through). To specify a bang path, you need to know the names of all the computers in the route. Don't worry about this too much, it's only here for illustration.

However, it does demonstrate the fact that the two systems are fundamentally incompatible. Software can be used to bodge-up links between systems running the two different types, but they're awkward to use as you have to specify two different addresses (one for internet and one for fido) and clumsy.

It also shows the difference in appearance between fido and internet addresses. The fido address tells you absolutely nothing about where or what that computer is, and because it's just a group of numbers, it's fairly hard to remember. But the internet address tells us several things: firstly, the .au at the end tells us it's in australia. The .org tells us the network it's part of is a non-profit organization. .apana tells us it's part of the australian public access network association. Of course, this last bit you wouldn't necessarily know, but all the rest follows reasonably straightforward rules.

When i refer to internet addresses and fido addresses here, i'm not talking about addresses *on* Internet or Fidonet, i'm talking about the style of addressing that those two systems use. There are hundreds of networks that use fido-protocol that aren't connected to Fidonet. This is probably the same for internet-protocol too.

Right, well, that all sounds a bit confusing, and in a way it is! Network addressing can be one of the most complicated aspects of computers. How complicated it is, depends on the size of the net you're dealing with. Also, fido addressing seems (to me, at least) considerably more complicated than internet addressing.

I know! I know!, i started off talking about fido and uucp and now i'm talking about fido and internet. Well, uucp is just one of the data transfer protocols that are used in internet. Uucp's the one that's used for batch network processing. The others

could be too, but they don't seem to be used that way yet.

Leaving the technicalities aside, the real differences between the two types of protocol - as they affect us if we want to start or join a network, are as follows.

The main advantage of fido protocol is that there are a lot of existing networks and bulletin boards that use it. If we want to link up with any of them, our bbs software must be able to do that type of message handling. The big disadvantage of this system is that we're going to have to find quite a bit of money to pay the phone bill.

The advantages of internet are that we can do our long distance links through internet for the cost of a local call. As well as this, we get access, via e-mail to a vast ammount of users all round the world. Also, addressing and routing of mail is considerably easier with internet.

The disadvantages are incompatibility with fido networks and not nearly as much bbs software easily available.

The main difference in terms of service provided by the different types of net is in the way what's known as <<echoes>> or <<conferences>> or <<newsgroups>> etc operate. These are public or semi-public message areas which carry messages around the net in a public way - as opposed to private email. In fido-type nets, you can set up these net-wide message areas easily, as you require them. But with internet it's not quite so simple.

Internet carries what's known as <<usenet newsgroups>> these are public areas where anyone can read or write messages on a particular topic. For instance: comp.bbs.waffle is a newsgroup where people discuss setting up and running a waffle bbs. However, not just anyone can set up one of these newsgroups. An alternative to newsgroups, which anyone *can* set up is a mailing list. To set up a mailing list, you have to be able to do that sort of thing on the computer you access internet through which means it's not really open to *everybody* to run one.

Anyway, the way a mailing list works is this: any mail sent to the list will be copied and sent out to everyone on the list. That way, the list's recipients can carry on semi-public conversations by email. This system isn't really as useful to us as the fido echomail system, however, by bodging things up a bit, it can be made to look similar.

8.2. Setting up a feed in a batch net

Whichever way you end up doing it, you'll still have to set up your system in the same basic way. Let's assume first off that you are joining an existing net, rather than starting one from scratch.

The first thing you have to do is find an entry point into the network. With fido nets this will probably be either the nearest node (bbs) in your country, or the nearest node in another country which can handle traffic to and from your bbs. It may well mean daily international phone calls (usually at a time when calls are cheapest) between your computer and your feed.

With internet protocol, networks can be set up in exactly the same way, with dial-up links between all nodes in the net. However, a more practical (and cheaper) way to do it is to get a uucp feed from an internet site in your local call area. Your computer then calls up the uucp site on a regular basis (at least once a day), passes all the outgoing mail to this system and picks up any incoming mail. The internet computer then automatically sends the mail off to its destinations.

Then, any other bbs in your net, anywhere in the world, can call up their uucp feeds and get your mail - and any other mail for them.

Organizing a uucp feed could be either very easy and cheap, or it could be difficult and/or expensive, depending on where you are in the world. Under some circumstances, you may have to make long-distance calls to your nearest uucp feed provider - then it will probably work out more expensive than if you were using fido protocol, as uucp seems to be slower.

Obviously, i can't give much information about how to find a uucp feed, as it varies from place to place. But the first

thing to look out for is a non-profit network of some sort. Going through commercial internet providers is not really a viable proposition unless you haven't got any other choice. If there isn't a public access, non-profit or community type of net where you are, you could consider trying to get one started up. Shared community resources are always much better than commercial ones and it shouldn't be impossible to get the support necessary to get a net together. However, that's likely to be somewhere a long way down the track from where you probably are if you're reading this, so to start with, you're going to be looking for an already existing gateway into the net. Ask around. And keep asking. There's so many little bits of internet scattered around and it can take a while to find them.

Check out the spunk press anarchist internet contact list for anyone in your part of the world who might be able to help.

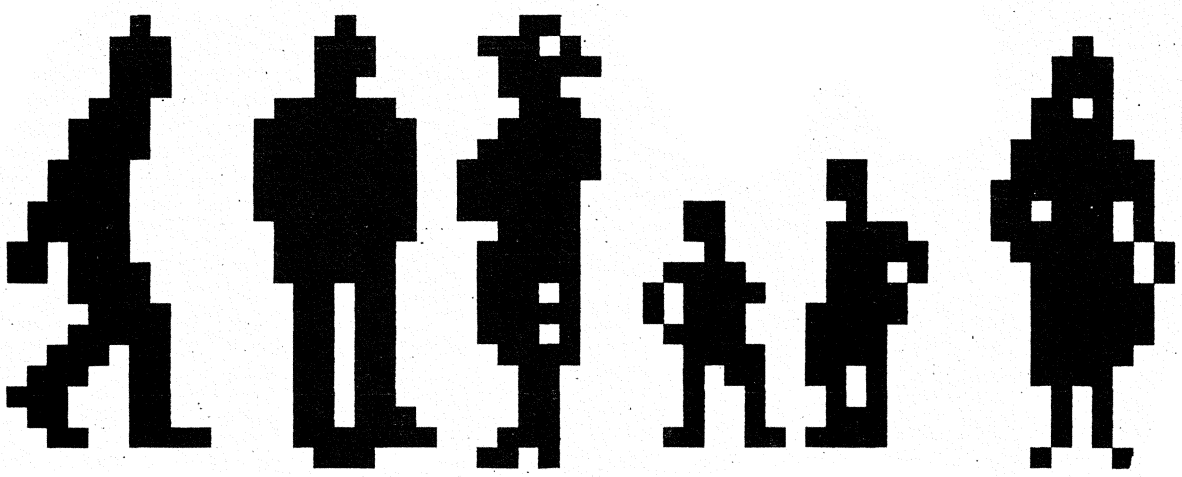
Once you've arranged your feed, you have to set up your bbs system to automatically poll the feed site. Polling means calling up and exchanging data.

Either your bulletin board software or your mailer software (e.g. front door, d'bridge etc) will have a scheduler (called <<event manager>> in front door) which you can set up to do certain things at preset times. You have to arrange a polling time (or, better, more than one polling time) with your feed site and then set up your scheduler to call it at that time.

I'm not going into all the technicalities of polling, as they're not only fairly complicated, but they vary from system to system. Anyway, this is one thing you'll have to sort out from the documentation that comes with your software. Don't forget, whoever's running your feed site will have experience in getting all this working and they should be happy to help you when you get stuck. And don't be put off by some of the weirdness of this aspect of networking, once you get the hang of it it's really quite simple.

One thing i must say though, is that when you're running a fido protocol system, you'll need three separate software packages: the bbs (e.g. superbbs, quickbbs, remote access), which does the things described in chapter 7, a mailer (e.g. front door, d'bridge, intermail), which answers the phone, manages events and does the polling; and a mail tosser (e.g. fmail, imail, tosscan), which sorts incoming and outgoing mail and makes sure they go to the right places.

With waffle (the only uucp bbs i know anything about), all these funtions are dealt with in the same package.



computer nerds on their afternoon off...

Chapter Nine

++++++++++++

9.1. Internet

There's so much hype about internet nowadays, it's almost impossible to grasp what it's all about unless you've experienced it yourself. And most of it *is* hype!

There's a lot of good things about internet, but if you listened to the hype merchants, you'd think the revolution had already come! It hasn't. There's a lot of talk about internet being anarchistic - which it is in a certain sort of way. But anarchistic as it may be, it's not anarchy!

For a start, almost all the major computers on the net (as well as the communication links between them are owned by the government - and most of those are owned by the u.s. government. The government internet computers that aren't owned by the military are mainly owned by universities. Almost all of the remaining computers that make up the main backbone of the internet are owned by large companies. Hardly my idea of anarchy!

Then, hanging off the basic backbone (i'm not necessarily using this term in the true net sense) there are a whole lot of little internet computers. A significant number of these are owned and operated either by non-profit or community groups or individuals who have a genuine desire to provide a community service. It's at this level that you can find the real anarchistic nature of the internet. However, without the government-owned and funded infrastructure, it would all be pretty useless - at least in the form it is today.

Apart from this, there isn't the worldwide revolution taking place on internet that the hypesters would have you believe. Sure, there's an incredible communications revolution happening here... But that's all. And it's hardly global. Try telling the majority of the world's population - who have barely got enough food, let alone access to computers, there's a global revolution happening on the internet! Access to a computer for most people in the world is about as far fetched as a flight to the moon, and access to internet is even further removed.

At the moment, internet's in it's infancy. It's anarchistic in its organization, in the sense of the way the net's connected, rather than the way it's run. It could be said that it's run by a benevolent oligarchy - a small group of people who allow things to run themselves. This gives users the illusion of some kind of autonomy. However, this illusion could be quite dangerous, as there's nothing at all to stop the government taking full control of what goes on on internet and ruthlessly suppressing the generally radical politics that govern the way it currently operates. This will most likely happen by upping the cost of access to drive off all the scumbags and turn it into a commercial venture. It's very unlikely they'll leave it to run the way it runs now for very long - it's too much of a threat to their power and to the monopoly the media barons currently hold.

However, bearing all that in mind, there is still plenty of good reasons to take notice of the increased availability of access to this massive international network. The main one being the possibility of very cheap communication between those people lucky enough to be able to get onto the net. This is rapidly coming to mean, most people in australia, europe and north america, as well as a few other people scattered around the world. To say these people are able to get onto the net doesn't mean it's easy for them to do so, just that the possibility's there.

So even though this cheap form of international communication isn't likely to remain within our grasp for very long, it's certainly worth making use of it now, so we can build some international links that would otherwise be extremely difficult. Once these links are built, they can be maintained even if we lose access to this channel. What we can do on internet today, we'll still be able to do with batch-type networks tomorrow, it will just be harder to initially make the connections.

9.2. What is it?

Internet is a realtime, international computer network. In fact, it's really a network of networks. This means, there's a lot of smaller, basically autonomous nets which are all linked together in some way to make up the whole thing. The Internet itself was, or is, a particular network within this large jumble, however, it's name has now come to mean the whole thing.

Generally speaking, anyone with access to internet will have an internet address. This is usually in the form: <username>@<domain>where <username> is one word with no spaces and can be a first name (e.g. paula), a composite name (e.g. sen.cheong), a group name (e.g. brisanarch), or anything else at all really. The <domain> part of the address identifies the particular computer that that user has an account on and could be something like murder.killnet.com.au - this type of address was explained briefly in chapter 8.

So anyone who has access to internet should be able to send mail to anyone else who has an internet address. No fucking around, just put their email address on the message and off it goes, automagically finding it's way to the right place. Email address means the type of address i talked about above. Email means electronic mail - that is, a message sent through a computer system or network.

Email is probably the most used service available to internet users. However, there are a couple more that are important to us: these are FTP and usenet newsgroups.

FTP stands for <<file transfer protocol>> and it allows you to get files from all round the world very easily. A large number of internet computers have public file areas and allow anyone to get copies of the files using ftp. The main types of file available, which are of interest to us, are software and text files.

If you've got access to ftp, you've got a massive international archive at your fingertips. Material on pretty well any subject you can imagine is there for the grabbing - finding where it is, of course, is another story altogether.

Usenet newsgroups are public, net-wide message areas. Each area has its own particular topic and other users won't appreciate you going too far off that topic. All messages on each newsgroup are available for anyone to read. And you can post your own messages there for others to read. These are basically the same as fido-type network echomail areas (described in chapter 8). There are thousands of usenet newsgroups (i don't know how many exactly) and how you find out which ones you're likely to be interested in, i'm fucked if i know! Spunk press contacts list includes a few of them which anarchists might find of interest.

Another internet service which will probably be of much less use to you, is called <<telnet>>. Telnet allows you to log on to any computer on the net which you've got an account on, or which allows public access in this way. It's more or less the same as calling up that computer over the phone, but it goes through the net, so it only costs you a call to your internet provider. Because your input and the remote computer's responses may go through many other computers and over lots of cables on the way, it can be an extremely slow process. The main uses for this service are to connect with your usual internet computer when you're off travelling or something, and connecting with one of the internet services which you use through telnet (e.g. archie - see later for explanation of this). You're probably not likely to use this much.

It's not really possible to explain how to use and access these services, as there's a lot of different software kicking about which gives access to them and each internet computer's likely to be different. However you can find out this information from the people running the system, other people who use the same system or one of the many guides to internet that are appearing in vast numbers almost daily.

It's important to be aware, though, that there's usually lots of different bits of software which do the same thing on each public access system. If the one you're using to read and write mail, or access

newsgroups etc, is a pain in the arse to use, it's pretty likely there's something better and easier to use available - you just have to know the command to use to get it happening. Ask system administrators for more information.

9.3. FTP

As i said before, with ftp you can get files from all over the world with a minimum or fuss. There are some very interesting text files around the place. An example is the spunk press archive of anarchist literature. This is located at the following internet address:

etext.archive.umich.edu

and can be found in the directory:

/pub/Politics/Spunk

(note the capitals here! internet computers mainly use an operating system called <<unix>>, which differentiates between upper case and lower case letters. Use the right ones or you'll come up against a blank wall!)

Note here, the address of the archive. Etext is the name of the computer it's actually on. This is (presumably) connected to a computer called archive which is at university of michigan (umich) and comes under the education domain (edu). In this type of address, there's nothing to indicate which country it's in. Often this means it's in the united states (of course! where else! it *is* the centre of the universe, after all!!!) but this isn't guaranteed.

Under the subdirectory Spunk there are a lot more subdirectories with all the text files in them. It's relatively easy to move around directories with ftp, using the command:

cd <dirname>

where <dirname>'s the name of the directory you want to go to.

An example of an ftp session to get the spunk press catalogue follows:

First, you type:

ftp

to get the ftp program running. Then type:

open etext.archive.umich.edu

which gets ftp to connect you to that computer. Then, once you've got a response back from the etext computer which indicates you're connected (this will almost certainly be followed by whatever prompt the ftp program uses when it's waiting for a command), you type:

cd pub/Politics/Spunk

when ftp indicates this is done (sometimes it takes a long time to get these responses back - they often have to travel many thousands of miles) you could then type:

dir

this will eventually display a listing of all the files in the directory **/Spunk** This will help you to get an idea of what subdirectories there are, etc.

Next you'd type:

cd contents

then:

dir

this will show you what's in that directory, but you don't *have* to do it. Then type:

get Catalog.txt

ftp will then transfer a copy of the catalogue to your home directory on the computer you're getting internet access through. You could then type:

quit

to end your ftp session.

Now, you could have done this in a couple of other ways. For example, if you'd started up ftp with the command:

ftp ftp://etext.archive.umich.edu/pub/Politics/Spunk/contents

(with no new line between **pub/** and **Politics/**) you would have gone straight into the directory you wanted. Then you could have simply entered the **get** command straight away. The problem with this method is that long string of characters that makes up the command above is quite difficult to type in all in one go without making a mistake. And if you make one mistake in the whole thing you'll be back where you started! I usually go through it in stages, like i've explained above.

9.4. URLs

The part of the above command which looks like this:

ftp://etext.archive.umich.edu/pub/Politics/Spunk/contents

is known as a URL. URL stands for uniform resource locator, and is an internet address with a particular format.

The first bit: **ftp://** shows it's the address of an ftp file or directory. The other type of url you're likely to come across is: **http://** this type is a world wide web address (more on this later). I'm afraid i haven't got a clue whether or not there are any other types of url.

Urls are a standard way to refer to files and directories on any computer on internet.

9.5. ARCHIE

There's millions of files publically available on thousands of internet computers around the world. A lot of them will be things that you either need (like certain bits of software) or are interested in (like text files). But finding what's where is a bigger task than actually getting the file once you've found it.

Archie is a service provided by a few computers scattered throughout the internet. All public access ftp sites (also known as anonymous ftp sites - because you can access them anonymously (that is you don't need an account on that particular computer)) regularly send lists of all their files to the archie servers. You can get archie to search through these lists and tell you if it finds the file you're interested in or something similar.

To use this, you have to telnet to your nearest archie site. The one in australia is:

archie.au

but there are other ones around the world - you'll have to find out yourself where your nearest one is. The command (for the australian server - and please don't use this if there's another one nearer you, the cross-pacific link is clogged enough as it is) is:

telnet archie.au

then you follow the instructions you're given once you're connected.

To get archie to do what you want, there's anumber of commands available. But the most important one is:

prog <string>

where <string> is the file name, or part of it, that you want to find. It takes a while for archie to search through the list, but eventually you'll get a (possibly quite long) list of sites where you can find what you're looking for.

You can also access archie by email. Send a message to archie@archie.whatever.your.local.one.is with **help** in the message body and you'll get an email message back with a list of the commands available.

9.6. Access to internet

Well, it's all very well to tell you about what you can do on internet, but i haven't said anything about how to get onto it...

Pretty much the same applies to getting interactive access to internet as it does to getting a uucp feed (see chapter 8). By interactive access, i mean you (probably) use a computer and a modem at home to dial up a public access site - the internet equivalent of a bbs, which you can get access to the net through. Another common way of getting internet access is through university, if you're a student or lecturer, in which case, you might use a computer terminal in the university itself.

Finding a public access site is obviously going to depend on where you live. In australia, apana (australian public access network association) run a non-profit public access network. There are a number of public access sites connected to this net, mainly in the capital cities on the east coast. For more information on apana, write to:

reply paid 7
propaganda dept
apana inc
p.o. box 145
keilor
vic 3036

or send email to:

propaganda@apana.org.au

include a daytime phone number so they can ring you.

Also, if you're outside of the main population centres in australia, probably the best source of internet connection is pegasus. They're expensive, but you do all your connection for the cost of a local call. I'm afraid i don't have their address handy.

However, i can offer no help to people in other countries. You'll just have to ask around. Don't necessarily go for the first one you come across - there's often an amazing difference in costs from one service provider to another.

9.7. Using internet

Once you're found a connection, you get access to all internet services by connecting to that computer through a modem and phone line. This way, your computer becomes a terminal of the internet computer and through it you can use the facilities available on that computer.

It's sometimes hard for people to grasp what's actually happening on their computer, what's happening on the internet access computer and what's happening on other computers they might be using (for instance through ftp).

When you use ftp, you're opening a communication channel from the internet access computer to the ftp computer (where the file you want is). The ftp computer responds to commands you give it through the access computer (and via your computer!!!) It then sends the file you want to the access computer, where it ends up in your home directory. Your home directory is an area of the computer's disk space which you can use to keep files and mail etc. Once the file you want is in your home directory, then you've got to get it down the phone line to your computer at home (or, hopefully, in your local infoshop or community centre).

To transfer files between your computer and the internet access computer, you usually use facilities provided by the communications software you're running (for example, zmodem). So once you've got the ftp file in your home directory on the access computer, you then have to download it to your computer. This is probably done with the command:

sz <filename>

Anyway, that's something else you'll have to work out yourself, as there's lots of differences between different setups.

One last point: your internet access computer may have user-friendly menus (like a bbs), but it might not. It's quite likely you'll just be thrown into a totally unhelpful, fairly blank screen. If you've used DOS,

you'll probably have a fair idea how this sort of thing works - if you haven't you'll probably be quite lost.

Most internet computers use an operating system called <<unix>> or another one which is very similar, called <<linux>>. These are similar to dos in general approach, but have a lot of different commands. It's a good idea to try and find a book about internet in your local library which will explain a bit about this stuff. It's also a good idea to learn a bit about dos. It will come in very handy. Windows is very pretty, but it doesn't help you learn how to deal with other environments.

Chapter Ten

++++++++++

10.1. Security

The term security has a few different shades of meaning when it's applied to computer stuff. The obvious use is in relation to keeping private things private.

10.2. Network Privacy

Computer communication is about as private as telephone calls. In other words, any given message might be private, but it could easily be seen by many different eyes along the way. When you're speaking on the telephone, there's always the chance that an engineer is listening to your conversation at any one of the (possibly dozens of) exchanges along the way. This is normal and they may well not be doing it from choice or noseyness - it's simply standard operational procedure for certain purposes. (However, if it's a particularly juicy conversation, there's a good chance they'll relay it over loudspeakers so all the staff in the exchange can have a laugh!)

Email through networks goes through a similar sort of system and may pass through quite a few computers on the way to where it's going. The sysops of any system on the way can look at your message - although it's not quite such a normal procedure as listening to phone lines.

So, although you can be reasonably confident that most of your mail won't be read, you must assume it can be under all circumstances.

This of course is quite separate from the other possible way your mail could be intercepted - surveillance by police or security services. Computer communication is no more or less susceptible to taps and electronic surveillance than phone calls. You should never say anything on any phone anywhere in the world that you don't want the police or government to hear. Likewise, you should never send any messsage over any computer network that you don't want them to read.

With phone calls, it's possible to scramble them so it's really difficult, if not impossible, for anyone else to decipher what you say. The same goes for computer messages, but it's easier.

To scramble phone calls, you need special equipment, which is expensive. But with computer communication, all you need is some software, which runs on the computer you're already using. The process is called encryption.

Data encryption software is fairly easy to get hold of. The only software i'd recommend is called PGP. Everything else is likely to be suss! But even PGP won't guarantee that someone who's determined to find out what you're saying can't decipher it. It will however take them time and a lot of extra effort - so it's worth using just to fuck up anyone who's that interested in your communications. But still, don't say anything you could go to prison for!!! Data encryption can make you feel your messages are secure - but you can't be so sure...

I'm not going to go into encryption in any depth. Just get yourself a copy of pgp somehow - it's available for several different types of computer - and read the documentation that comes with it. It's a bit confusing in places, but it's not as complicated as the manual makes it seem, and there's a good rave on encryption and political stuff etc...

The main thing to remember about security in this context is: assume that everything you send around a network can be read by anyone who's interested enough to go to the hassle it involves for them. It's not much hassle for system operators if your mail isn't encrypted - it's pretty impossible if it is.

It's a fair bit of hassle for cops if it isn't encrypted - and a lot more hassle if it is. Whether they'll bother or not depends on two things: firstly, whether they think it's worth what they'll get out of it - and when it comes to special branch and asio, who knows what they'll consider is worth it, they're really weird! Secondly, particularly with political cops (asio and special branches) it depends on whether there's

anyone else more interesting to watch... In australia, there's virtually nothing of any politically dodgy importance going on, but they've still got to justify their existence. This means, they're probably tapping your phone (and therefore your data communications). I know you're not doing anything very exciting, but no-one else is either and they've got to tap somebody's phones, so why not yours?

If you're planning to do anything you wouldn't talk about at the reception desk of your local police station, don't talk about it on (or even near) the phone or over a computer. Face to face in the middle of a park is better.

10.3. BBS Security

The other main meaning the word <<security>> has in computer contexts is in relation to who's allowed access to your computer files or message areas on a bulletin board etc.

All bulletin board software has a means of allocating all users security levels. All areas of the bulletin board can be set up to have security levels too. This means that anyone with a personal security level less than the security level of (say) a particular files area, can't look at the files in that area - they probably can't even tell that area exists. The same goes for message areas.

This is obviously important for more than one different reason. Firstly, you might have message areas where particular collectives discuss collective business - you don't necessarily want outsiders joining in these discussions or checking out what they're about.

Also, you could have some sensitive or even possibly illegal files on your system that you want to restrict access to for obvious reasons (like maybe you don't want to get busted...)

Most bulletin board software will allow you to restrict access to certain areas in other ways too: security levels are a bit crude for some things. You can usually give people access to certain groups - either by setting flags (like in superbbs) or by giving them access to numbered groups (like in waffle). This means that people with the same security levels can have access to different discussions or file areas. It's possible you won't use this stuff much, but it's worth being aware it's there if you need it.

The other aspect of this which is important is if you're setting up a public access computer in a bookshop, infoshop, community centre etc. If you want to run a public access bulletin board system in this way, there's certain steps you'll need to take to keep it secure.

The main thing is you don't want to let just any old body have access to your bbs computer. The only access they should have to this system is the same access they'd have if they were calling in over a phone line. This means they have to use the bbs as if they were - in other words, you must provide a second computer to act as a public terminal for the bbs. This can be any old shitty, cheap, nasty, slow old heap that can run a communications program and has a serial port on it.

You can connect the terminal to the bbs computer through a null modem and let people do their bbs stuff through that. The bbs computer can be locked in a cupboard or have the keyboard locked -either physically, with a key, or by software (such as frontdoor, which has this capability). Also, it should probably have the video display turned off or disconnected while it's not being used by sysops. This is specially true for waffle, which displays the users' passwords when they enter them!

If you've got a dial-up line running on the same bbs, you'll have to either set it up as a multiline bbs, with the null modem one one line and the phone line on the other. Or you can set up two computers with the bbs runing separately on both, sharing files over a lan (local area network) and a third one acting as the public access terminal. This last method is better if you've got access to a few shitty old XTs (say) which you can't do multitasking on (and therefore can't set up a multiline bbs on).

10.4. False Names

Usually known as aliases, these really are the normal way of identifying yourself in the BBS world. They don't seem to be used nearly so much in the internet world though. Most people on internet use their real names - this is probably because of the difference in the way the two systems are used.

I think it's safe to say that on bbss the majority of messaging is done via echomail (i.e., basically public message areas) whereas on internet, it's mainly done through personal email. If you change your name on a bbs, chances are you'll still be able to read any messages addressed to your old personality. This doesn't apply to internet - except for usenet newsgroups, of course.

Also, of course, everyone on internet has their own email address that is much more connected to them than the equivalent on bbs networks. The software's designed so any message you send has your email address at the top. It's not very easy to change your email address.

There does exist what's known as anonymous email services on internet, which compensate for this. They are run by individuals as a public service and operate by switching your name at the top of your message to a false one and inserting the anonymous mail service's email address in place of yours. They keep track of who everyone is and forward any replies. Of course, these aren't necessarily to be trusted fully, but they are useful.

There are certain advantages to anonymity when it comes to net communications, but there are also certain disadvantages. The main one is that if you're communicating with people you know in real life how are they going to keep track of who's who? For instance, if we set up an anarchist computer network in australia, we're going to be mainly communicating with people we already know personally. If we all use false names, it's going to be impossible to work out what's going on. And it *is* important to know who's talking about certain things. Often the way we react to ideas depends on who they're coming from - sometimes this is a bad thing, but just as often it's a good thing too. We know who we can trust and who we can't, and covering our real identities with false names is going to make it harder to work together.

I think it's best generally to use your real name (or at least a close approximation) - after all, if it's the cops and asio you're hiding from, don't bother, they can work out who you are anyway. Just never do or say anything illegal using your real name. False names can come in handy occasionally for such things, but don't let them give you a false sense of security.

Anyway, having said all this, i must make it clear that these are just *my* opinions. Particularly in the area of security, i don't want to tell anyone what they should do. You have to work out for yourself what's the best approach and that way, it's your responsibility and no-one else's. All i can do is give as much information as possible - but don't forget, i could easily be wrong.

10.5. Data Safety

It's debatable whether this should come under security or not, but protecting your data from loss or accidental destruction is an important thing put a bit of energy into. It's very easy to accidentally erase a file and not be able to un-erase it. It's also not uncommon for component breakdowns (most notably hard drives) to completely trash everything you've built up over months or years.

The only way to protect against this is to keep copies of everything that you can't really easily get again. This process is known as <<backing up>> or <<making backups>>.

The proper way to make backups is to always keep multiple copies of everything. The first time you back up your files, you'll only have one copy. But the second time you do it, you should keep the first one as well - as it's not impossible to accidentally end up with an unusable backup too! Then

the third time you do it, keep the first two. After that, re-use the disk(s) you used for backup number one. Then overwrite backup number two etc. Keep going round in a cycle this way, making new backups after every significant change in your files.

If you follow this advice you'll never know how important it is to do so! You'll be one of the lucky ones (do they exist?) who don't tear out their hair every time something fucks up and they loose everything on their systems. Starting again from scratch, with no backups can easily be enough to stop you in your tracks and make you take up tiddlywinks instead. It's much better *not* to learn this through experience!!!

SPUNK PRESS MANIFESTO

The excuse for the existence of Spunk Press is the desire of some individuals to see alternative literature continue to flourish, but this time online!

The policy of Spunk Press is to act as an independent publisher of works converted to, or produced in, electronic format and to spread them as far as possible on the Internet and in the BBS society free of charge. The work may not necessarily originate from someone with net access. The major interest of Spunk Press is alternative literature and anarchist material, both old, converted, and newly produced.

We want to help fanzine editors, flypost authors and others who desire a wider audience to convert or to produce their works in an electronic format and give them the opportunity to use our distribution channels, FTP sites, mailing lists and whatever other means we might have within our powers.

We welcome fanzines, pamphlets, books and portions of books, articles, manifestos, quotations, interviews, bibliographies, reviews, posters, and other material, both in-print and out-of-print. To contribute with material, contact the editorial collec-tive.

Spunk Press is run by an editorial collective. The collective has access to a mailing list forum, open for everyone, for decision making. If there is no clear consensus on the forum, or the consensus is at variance with anarchist ideas, the collective decides. The collective is composed of people with a reasonable commitment to doing some aspect of the work at Spunk Press, and will be extended to those who are like-minded.

If you would like to receive regular information about Spunk Press, send a message to:

Spunk Press Information <spunk-info-request@lysator.liu.se>

If you would like to reach the editorial collective of Spunk Press, send a message to:

Spunk Press editorial collective <spunk@lysator.liu.se>

Spunk Press Scotland

—Ian Heavens
ian@spider.co.uk

Spider Software
Spider Park,
Stanwell Street
Edinburgh, EH6 5NG, Scotland
+44 31 555 5166 (Ext 4735)

Spunk Press Manifesto Version 2.2, June 1 1994

Index

A

address 22
anonymous email 33
anonymous ftp 28
archie 26
ascii 7
asio 31

B

backing up 33
bang path 22
batch 18
baud 11
bbs - see bulletin boards
binary 8
bit 7
bps 11
bulletin boards 10, 13
byte 7

C

communication 10
computer 3
configuration 5
cp/m 6
cps 12

D

data bits 12
dodgy 32
domain 26
dos 6

E

e-mail 23
echoes 23
encryption 31

F

female connectors 10
file names 8
floppy drives 3
ftp 26
fucking 26

H

hard disk drive 4
hayes 12
help 15
hexadecimal 8

I

initialization 12

L

lan 18
linux 6
logoff 14

M

macintosh 6
mainframe 22
male connectors 10
minicomputers 22
modem 10
mother board 4
multi-tasking 6

N

network 18
null modem 11

O

operating system 5
option cards 3
os/2 6

P

parallel 10
parity 12
pdp11 22
pdp8 22
pgp 31
police 31
ports 10
prison 31
processor 5
program 5
protocol 12

R

realtime 18

S

security 32
serial 10
serial card 10
software 5
special branch 31
speed 11
spunk press 26, 35
stop bits 12
sysop 14

T

telnet 26

U

unix 6
url 28
usenet 23
uucp 22

W

wan 18
windows 6

The author, (above, with "close friend" Chi-chi), lives far from the censoring eyes of society, deep in the bush of northern New South Wales.

After a tortured childhood, with many short-term relationships, the most notable with "squeak", he retreated into the world of the computer, where he developed an extensive knowledge, offered to you here at a remarkably cheap price, on how to operate you own system, with maximum autonomy from the wiles of big business and government.

Chi-chi led to the end of his computer obsession and the start of a new life of candlelit dinners, single red roses and romancing under the moon at midnight.

SCAM publications

P O Box 756 Brunswick Lower Vic 3056

e-mail : scam@xchange.apana.org.au